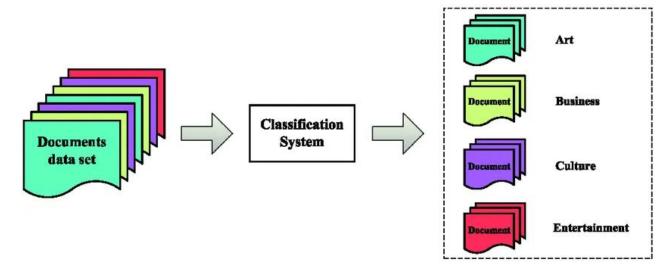
# Hebrew Articles Category Classification

Baraa Saleh - 206466732 Marah Aboud - 208865907

### **Business Model**

Our end goal is to create a classification model, that is able to classify articles into their respective category by leveraging data analytics and machine learning techniques

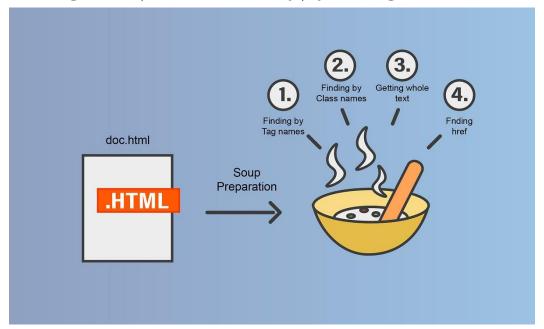


# Roadmap

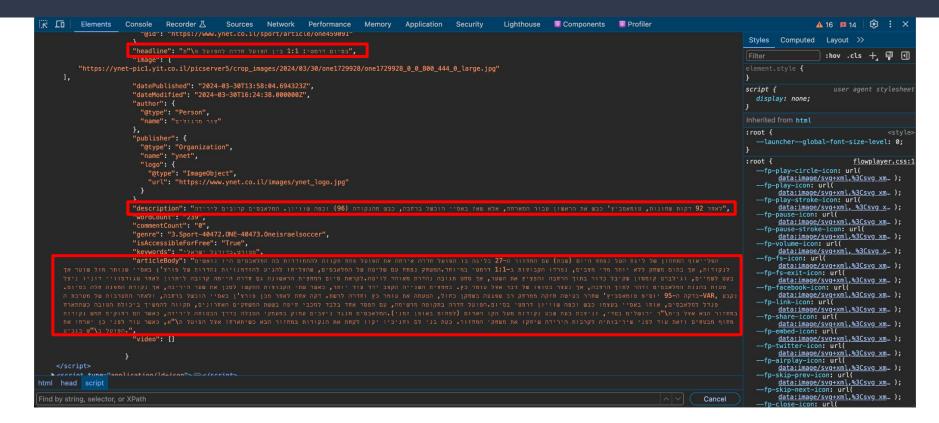
- Data scraping
- Preprocessing
- Classification model
- Backend: Flask
- Frontend: React

# Data Scraping: Ynet + BeautifulSoup

We used Ynet which is a popular website containing articles in hebrew and parse the HTML using bs4 (BeautifulSoup) package.

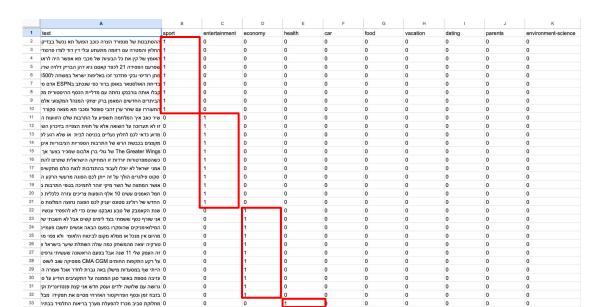


# Data Scraping: Headline + Description + Body



# Data Scraping: Save the data

Save the article text and its respective category into a Pandas DataFrame and then save the DataFrame into a csv.



# Preprocessing: NLP

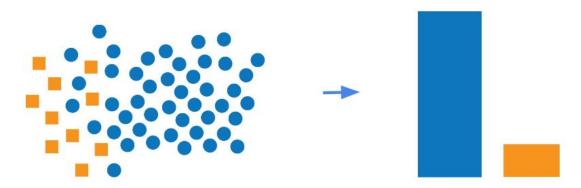
Analyze and clean the data using NLP techniques, such as removing the stop words from the texts which do not add to the understanding of the text.

For example: את, לא, של, אני, על, זה, עם, כל, הוא, אם, או, גם, ...

## Preprocessing: Imbalanced Data

While checking for imbalances in the data, we noticed that 8 categories had ~900 articles and for 2 categories there were < 50 articles so we removed these 2 categories from the data due to lack of support.

Final data size ~7000 articles



# Preprocessing: Encode the categories

Replace all the categories with an encoding to describe the category this text belongs to.

#### For example:

- 0 Sport
- 1 Entertainment
- 2 Economy

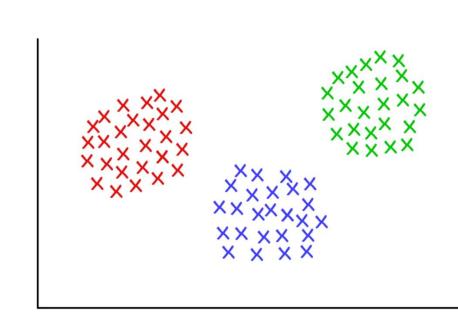
etc ...



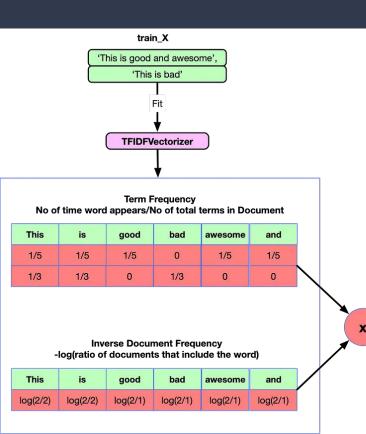
### Multiclass Classification

Classification is the problem of assigning observations to one or more categories.

Multiclass classification involves more than 2 categories.



#### TfidfVectorizer



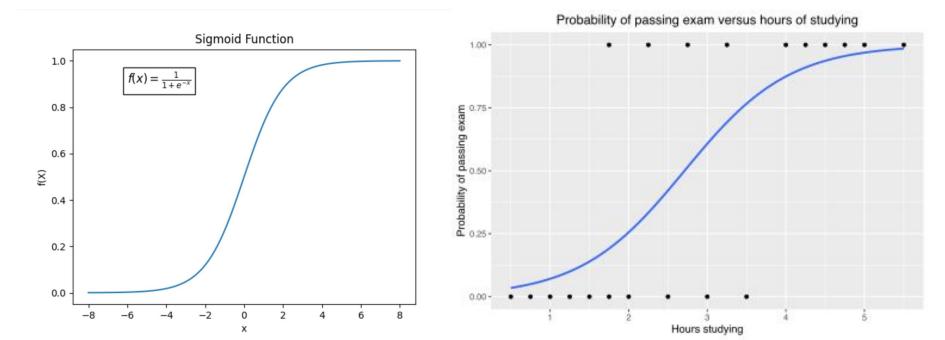
Convert our collection of texts to a matrix of TF-IDF features, which is a formula that aims to define the importance of a word within the text

	This	is	good	bad	awesome	and	
<b>-</b>	0	0	1/5*log(2/1)	0	1/5*log(2/1)	1/5*log(2/1)	
	0	0	0	1/3*log(2/1)	0	0	

Features

# Classification Model: LogisticRegression

LogisticRegression is a model based on sigmoid function, and here is an example of 2 classes:



# Classification Model: LogisticRegression

In our case, we have multiple classes (8 to be exact) and the graph looks similar to this:

Important to note that the sum of the probabilities of a single point is 1.

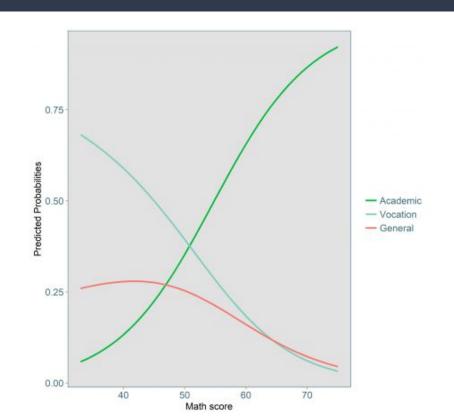
For example, if we look at grade 48, we can see that the probabilities are:

0.5 - Academic

0.25 - Vocation

0.25 - General

And their sum is 1



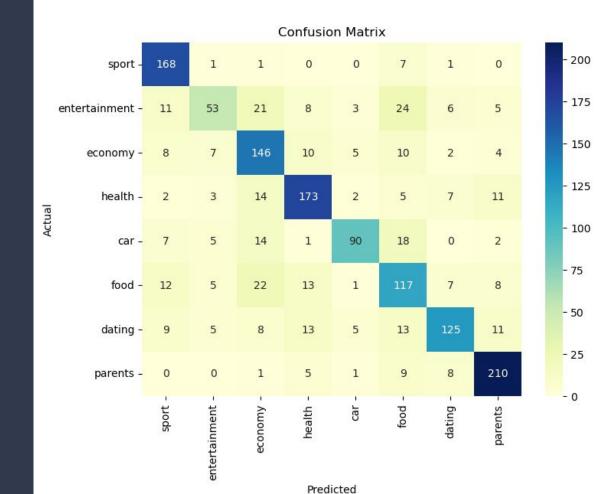
# Classification Model: LogisticRegression

#### Our train and test process:

- Vectorize the "text" column into vectors (this is the input data X)
- Split the X and y data into X\_train, X\_test, y\_train, y\_test using train\_test\_split function, where the test\_size is 20%
- Fit the model
- Save the model and the vectorizer into .pkl files

## Confusion Matrix

Testing the model on %20 of the data and see how many times the model answered correctly



### Model Performance Metrics

LogisticRegression									
	precision	recall	f1-score	support					
0	0.77	0.94	0.85	178					
1	0.67	0.40	0.50	131					
2	0.64	0.76	0.70	192					
3	0.78	0.80	0.79	217					
4	0.84	0.66	0.74	137					
5	0.58	0.63	0.60	185					
6	0.80	0.66	0.72	189					
7	0.84	0.90	0.87	234					
accuracy			0.74	1463					
macro avg	0.74	0.72	0.72	1463					
weighted avg	0.74	0.74	0.73	1463					

### Model Performance Metrics

Here are a few metrics that helps check model performance.

For example, let's look at the Sports category:

Precision = 168 / (168 + 49) = <u>**0.77**</u>

Recall = 168 / (168 + 10) = **0.94** 

0.85

F1 = (2 \* 0.77 \* 0.94) / (0.77 + 0.94) =

Accuracy for all = 1082 / 1463 = **0.74** 

 $Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$ 

 $Precision = \frac{TP}{TP + FP}$ 

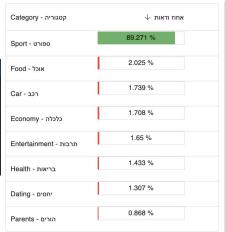
 $Recall = \frac{TP}{TP + FN}$ 

 $F_1 = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$ 

### Backend + Frontend

Backend: Flask server loads the model the vectorizer and listens for requests. Frontend: React framework, we send requests to the backend and show the results in a table.

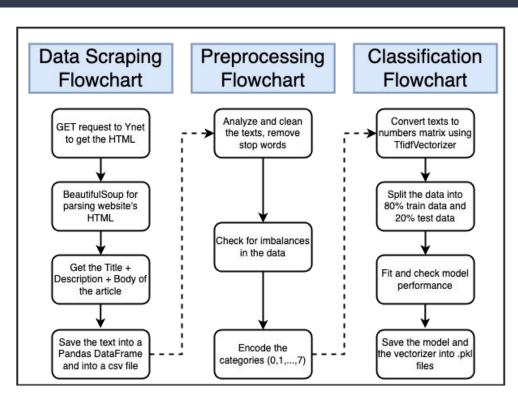


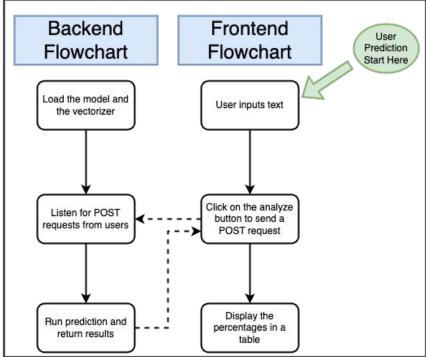


למורסיה: מרקו תודורוביץ' 23 נקודות, דסטין סליבה 13 נק', דילאן אניס 10 נק', ג'ונה ראדנבו 9 נק', מוסה דיאן ורודוניוס קורוקס 2 נק' ט"א וטרוי קופיין נקודה אחת.רבע ראשון - 16:17 למורסיהחמישיית הפועל חולון: ניב משגב, דרו קרופורד, סי ג'יי האריס, קווין הרווי וג'סטין סמית'.חמישיית מורסיה: טרוי קופיין, דילאן אניס, רודיונוס קורוקס, דסטין סליבה, ומרקו תודורוביץ'.סליבה וקורוקס היו אחראים לחמש הנקודות הראשונות של הספרדים, כשהרווי היה הראשון לעלות על הלוח בצד של חולון. שלשה של סי ג'יי האריס השוותה את התוצאה, אך אניס החזיר מיד עם אחת משלו. ניב משגב והרווי קלעו מחצי מרחק ושלשה של האריס העלה את החבורה של עמית שרף ליתרוו ראשון בהתמודדות.היתרון החל להתנדנד מצד לצד כשג'ונה ראדבו קלע מעבר לקשת מהפינה ובצד השני טאג'יר מקול דייק פעמיים מהקו. הגארד של חולוו נחסם במהלר הראשוו של הרבע ולוח התוצאות הראה 16:17 למורסיה לאחר 10 דקות של כדורסל.רבע שני - 31:33 למורסיהשחר עמיר קלע שלשה במהלר הראשוו של הרבע. מקול קלע גם כו בפתיחה כשבצד השני ארבע נקודות של באדבו קבעו 21:21. האנס ואנווין הוסיף שלשה ושתי נקודות של הרווי מהקו העלו את הסגולים ליתרון חמש. ההיילייט הגדול של הערב הגיע מניב משגב שהרים האלי הופ מחצי מגרש לג'סטין סמית' שסיים בדאנק. סיטו אולנסו עצר את הדהירה של חולון עם פסק זמו.סמית' עם סל ועבירה העלה את ההפרש לשמונה, תודורוביץ' קלע והוסיף נקודה מהקו ונקודות של אניס צימקו את ההפרש לשלוש בלבד. תודורוביץ' וסליבה השלימו ריצת 0:10 שהחזירה את היתרון לספרדים וגרמה לעמית שרף להזעיק פסק זמן. סמית' ומקול החטיאו זריקות פנויות מתחת הסל לקראת סיום הרבע ובסיומו הישראלים ירדו בפיגור 33:31 לאחר שלא קלעו נקודות בכמעט ארבע דקותיו האחרונות.רבע שלישי - 43:56 למורסיהלמרות נקודות של ניב משגב, פתיחת הרבע הייתה של הספרדים ושלשה של תודורוביץ' העלתה את היתרון שלהם לחמש הפרש. הסגולים ניצלו בזכות כמה החטאות קלות



#### Flowchart





### Technologies

- Data scraping: BeautifulSoup, Pandas
- Preprocessing: NLP, Imbalanced data, Encoding
- Classification model: TfidfVectorizer, LogisticRegression
- Backend: Flask
- Frontend: React



#### סיווג טקסטים לפי קטגוריות

